Rank Fusion
Information Retrieval Project

Davide Rigoni - Silvia Colucci - Alex Beccaro

January 12, 2018

Abstract— Recently researches have focused on the idea that performance and the effectiveness of individual IR models can be improved by combining the outputs of some individual models into a single result set. Data fusion is the combination of the results of independent searches on a document collection into one single output result set. In fact, ranking quality can be achieved by adopting different strategies and so different retrieval models. In this regard, we have implemented some rank fusion base strategies to perform retrieval experiments by generating runs and by comparing their individual results with results obtained by ten runs of different systems through Terrier. Then we have implemented an advanced rank fusion strategy called ProbFuse to perform another results comparison and to look for eventual improvements compared to the popular fusion algorithms.

I. INTRODUCTION

It has been shown in the past that rank fusion can greatly improve retrieval effectiveness over that of the individual results. This improvement can be achieved by evaluating the results obtained by implementation of different rank fusion algorithms and by the developing and comparing new rank fusion techniques. In this work ten runs are been generated through Terrier by choosing different IR models to catch all the important features from the given files collection. Then, through the implementation of the classical rank fusion algorithms (CombSUM, CombMAX, CombMNZ) we have evaluated the generated runs compared also with those obtained through Terrier previously. Finally, we have realized ProbFuse algorithm: an interesting method that could led us to the achievement of better IR results. ProbFuse's idea consists in ranking documents based on their probability of relevance to the given topic in a reference k segment. This probability depends on which input system returned the document in its results and the position in which it is returned in the result set. The inputs to the fusion process are produced by different IR models running the same topics on the same document collection.

II. COLLECTION FILE UNPACKING

At the beginning it has been difficult decompress the collection files. It has been discovered that Terrier's 4.1 version was bugged and so we have had to use the 4.2 version to unpacked all the files. The zip file to be decompressed needed to one decompression program but it contained other zipped files that needed to another decompression program, they were compressed by using Unix. Some file were named with their names followed by an integer number and their extension names, such as if they were divided in small groups. They needed to be renamed, so it has been written a bash file to specify how to rename and decompress them automatically.

III. INDEXING

It has been used Terrier to index Trec collection. At the beginning, it was unknown what was the just format and the exact markup structure to adopt for the given collection through Terrier, so it has been complicated to understand how to set up the properties file and create the indexing with the appropriate lexical analysis, stop-word removal, stemmer and the other settings. The stop-list and the stemmer used for indexing are those base of Terrier used for English language, the stemmer is PorterStemmer.

IV. CHOICE OF IR MODELS FOR RUN BEGETTING

The followed models are been chosen though Terrier to generate ten runs to evaluate subsequently:

1) TFIDF
2) BM25
3) DFRBM25 (DFR)
4) LGD (DFR)
5) PL2 (DFR)
6) InexpC2 (DFR)
7) InL2 (DFR)
8) DLH13 (DFR)
9) IFB2 (DFR)
10) BB2 (DFR)

These are very different models based in sundry approaches, some of them have something in common, other use different functions if we compare them to some others, so they are different each other generally. In this way, the rank fusion should achieve better results, because if it takes values from all the individual models, it can capture collection qualities and strength elements given by these and so the fusion should report good results.

V. NORMALIZATION

The normalization used in this work is been the Standard Normalization which calculates normalized scores by using the maximum and the minimum scores given in the result set. The minimum score is subtracted to the unnormalized score of each run element and then this result is divided for the difference between the maximum score and the minimum score. After the score normalization, the followed combination functions for the scores combination are been implemented: CombMAX, CombSUM, CombMNZ, ProbFuse. They allow to carry out the rank fusion.

VI. PROBE FUSE ALGORITHM

ProbFuse algorithm needs to build a set of probabilities for each input model. The analysis of the performance of each individual models on training queries allow to calculate these probabilities. In our case, we have only one query for the whole collection. To calculate these probabilities we need runs given by the individual models and we need to divide each topic of each run in k segments. For each segment we count relevant and not relevant documents and we calculate the probability that considers each relevant document d returned by query considering each topic. So, for each segment we want to calculate the probability that a relevant document returned in this segment is relevant to the given query. Each probability is obtained by the relationship between the number of relevant documents for a specific topic in a k segment and the number of all the documents inside the considered segment. Each element in the segment is identified by topic and document name, we assign its probability calculated in the segment and finally we have to sum the probability of the same documents in the different runs. Thus, the resulting elements will form the fused run by using the probabilities calculated. We have considered segments of fifteen, twenty, thirty and fifty size. Then, in the evaluation phase it has been discovered that we can obtain the best results by considering segments of fifteen size.

VII. IMPLEMENTATION ISSUES

At first, it occurred a lot of time to execute main, the reading phase was very slow; the slowness was due to the structure used for Standard Rank Fusion functions. This problem has been solved by changing the structure from List to HashMap. In these functions the research by score of run elements needed to iterate all run elements list, so complexity was O(n). Instead, by introducing HashMap structure the access for example to the element with maximum score value can occur directly. In this way the complexity is reduced to O(1) and main program can be executed in few minutes. During the printing phase on file there was another problem: during this operation, after the execution of for cycle that iterates inside the run, it was created a single string. It took run's rows and created a single string that was passed to an object which wrote a single string on file, but the string object was immutable and so every time it was created a new object copied by precedent string that added the new row. This caused copy of every run rows. This has been solved by using an object able to write on file through enquequing elements inside a buffer, so that an element can be enqueued after the previous, the copy is avoided and the problem disappears.

VIII. RUN ASSESSMENT

To evaluate all the strategies adopted in this work it has been used Matters library to calculate all the necessary assessment measures. The principal measures performed for the comparisons are been: the calculation of precision in different levels of cut-off, RPrec, Average Precision, Cumulative Gain and Discounted Cumulative Gain. Precision in different levels of cut-off is been calculated because it is very important to mark runs that have relevant document in high position, the ten and thirty cut-off levels are been considered. If we consider cut-off level equal to the recall base (the number of the relevant documents), we can obtain another interesting measure that is R-Prec. Instead, Average Precision is the most important measure used because it gives power to relevant document present in high position rank. DCG is used also to consider position of relevant documents, the discounted function decreases the document weight in the bottom ranking levels. It is calculated at different cut-off levels but it has not much meaning because we can't know the maximum possible value that changes in the topics so it has been considered less than the other measures. By comparing measures between runs evaluated by Terrier and those returned by standard rank fusion implemented algorithms, it has been noticed that on average the values of the measures of the different strategies are very similar. Higher values can be obtained through rank fusion standard methods for some topics, but there are also some lower values for other topics compared to runs generated by Terrier. This difference is very small so it can be considered negligible. We have reported some examples where results obtained through standard stategies are better than runs evaluated by Terrier.

TABLE I
COMPARISON BETWEEN RPREC VALUES

T	ComMNZ	ComMAX	ComSUM	PL2	LGD
351	0.41667	0.41667	0.41667	0.375	0.45833
353	0.46721	0.31148	0.46721	0.30328	0.45082
396	0.33898	0.33898	0.33898	0.27119	0.33898

TABLE II
COMPARISON BETWEEN AVERAGE PRECISION VALUES

T	ComMNZ	ComMAX	ComSUM	PL2	LGD
351	0.42182	0.41482	0.42236	0.36367	0.45134
353	0.2235	0.16493	0.22377	0.16542	0.23566
396	0.29961	0.3315	0.29961	0.27036	0.33576

By calculating the assessment measures of rank fusion obtained with ProbFuse it can be noticed that all the values aren't very similar. For about ten topic, the results achieved are worse than standard rank fusion algorithms, but for the remaining topics, on the average, the values are similar. For some topics we can achieve even better results than standard rank fusion strategies. In particular, for the followed topics we have noticed improvements in ProbFuse assessment results: 355 and 359 topics (in precision measure at thirty cut-off considered level), 394 topic (in Rprec measure), 399 topic (in precision measure at ten cut-off considered level), 363, 367, 369, 370, 374, 376, 378, 379, 398, 380 topics in all the assessment measures. We have reported small examples where ProbFuse results are better than those obtained through standard strategies.

TABLE III
COMPARISON BETWEEN AVERAGE PRECISION VALUES

T	ComMNZ	ComMAX	ComSUM	ProbFuse
363	0.047316	0.039032	0.047187	0.13522
378	0.0057959	0.004285	0.0057962	0.021935
380	0.17362	0.10718	0.17408	0.25727

TABLE IV
COMPARISON BETWEEN RPREC VALUES

T	ComMNZ	ComMAX	ComSUM	ProbFuse
363	0.125	0.0625	0.125	0.25
378	0.14286	0.14286	0.14286	0.28571
394	0.058824	0	0.058824	0.11765

TABLE V
COMPARISON BETWEEN PRECISION VALUES AT TEN CUT-OFF LEVEL

T	ComMNZ	ComMAX	ComSUM	ProbFuse
363	0.2	0.1	0.2	0.25
378	0.4	0.4	0.4	0.6
394	0.2	0.2	0.2	0.6

TABLE VI
COMPARISON BETWEEN PRECISION VALUES AT THIRTY CUT-OFF LEVEL

T	ComMNZ	ComMAX	ComSUM	ProbFuse
359	0.033333	0.033333	0.033333	0.16667
370	0.1	0.13333	0.1	0.33333
379	0.033333	0.033333	0.033333	0.1

IX. CONCLUSION

We have implemented standard rank fusion strategies and ProbFuse algorithm, a data fusion algorithm that relies on the probability of relevance to calculate a ranking score for documents in a fused result set. These probabilities are calculated based on the position of relevant documents in result sets returned in response to the given collection's query. For this collection, full relevance judgments were available, so all the document were just been judged. By assessing the fused result set given by the different strategies adopted, we have discovered that standard rank fusion strategies are able to achieve results very similar to those obtained through fused result set derived from runs given by different individual models evaluated by Terrier. For some topics, the use of standard strategies has led to achieve results even better than these last. Instead, by comparing fused result set obtained through ProbFuse algorithm and those of standard strategies we have obtained better results for some topics that are been cited previously, for many others topics we have obtained similar results and for about ten topics the results were worse than those derived from standard rank fusion strategies.